Course Title: High Performance Computing

Credit Structure (L-T-P-Cr): (3-0-3-4.5)

Course Code: CS301

Prerequisites (if any)/desired skill set: Good background in C programming and Linux

Course objective: This course aims to provide a systems perspective towards achieving the maximum possible performance out of a particular computing system for a particular algorithm/problem. This course is an introduction to parallel computing and aims at teaching basic models of parallel programming including the principles of parallel algorithm design, modern processor and parallel computer architectures, performance considerations, programming models for shared and distributed-memory systems, message passing programming models used for cluster computing etc. as well as design of some important scientific and engineering algorithms for parallel systems.

Course content:

Multi-threading model using OpenMP. OpenMP: parallel do, private variables, nested loops, reductions, loop dependencies, thread-safe functions, parallel sections, and barriers. Message Passing Programming, its implementation and important details, MPI send and receive, MPI communicators, broadcast, reduce.

Parallel complexity analysis, speedup, efficiency, cost, task and data dependency graphs. Several Important Parallel Algorithms and implementation strategies from different class of problems such as Integration using trapezoidal rule. Vector addition, Calculation of PI using monte carlo method. Matrix operations. Reduction, Inclusive and exclusive scan. Image processing. Sorting algorithms. Solution of Differential Eqns using Finite Difference etc. Hybrid parallelization with MPI and OpenMP.

Labs: three hours per week. Major part of the course includes Lab Component for actual implementation after learning the basics of Parallel programming and HPC.

Project: Compulsory two-month group project and project presentation.

Course Outcome: Ability to design and implement of parallel algorithms on shared and distributed memory architectures. Code optimization and performance analysis of parallel codes.
**Evaluation:**

Midsem exam: 30%
End Sem: 35%
Lab Assignments: 15%
Course Project: 20%

Grading scheme is relative and depends on both: class performance and minimum expectation from a student.

**Suggested textbook/references:**

*Let's HPC: A web-based platform to aid parallel, distributed and high performance computing education* ([https://doi.org/10.1016/j.jpdc.2018.03.001](https://doi.org/10.1016/j.jpdc.2018.03.001))

1. An Introduction to Parallel Programming; Elsevier; by Peter S. Pacheco.
2. Scientific Parallel Computing; Princeton University Press; by Babak Bagheri Terry Clark L Ridgway Scott Bagheri Clark Scott
3. PARALLEL PROGRAMMING; Barry Wilkinson, Pearson Education.
6. Parallel Programming in C with MPI and OpenMP; by Michael J. Quinn; McGraw-Hill Higher Education